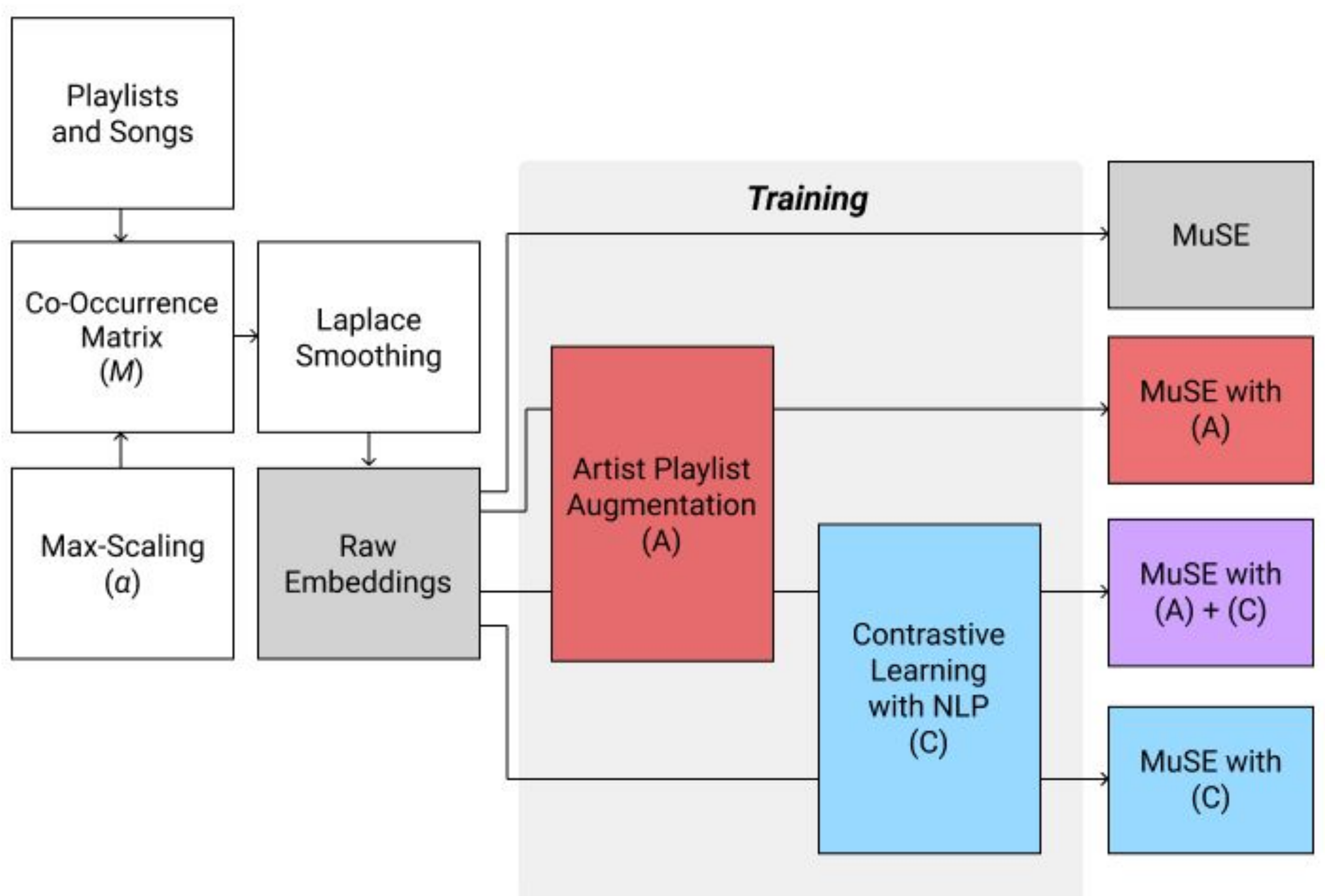
Musical Semantic Embeddings (MuSE)

Eric Lee and Akshar Sarvesh
{ericlee7, asarvesh}@stanford.edu

Stanford
Computer Science

Motivation

- With the rise of personalized music streaming platforms like Spotify and Apple Music, song recommendation systems have become integral to user experience.
- Current recommendation systems are often computationally expensive, requiring complex models and extensive feature extraction for each individual song.
- We introduce Musical Semantic Embeddings (MuSE), a novel approach inspired by GloVe, using only playlist data to generate robust song embeddings for large-scale recommendation.



Data and Features

- We sampled **527** playlists containing **21,747** unique songs from a Spotify Playlists data set.
- The dataset includes information about playlist names, song names, song artists, and maps each playlist to the list of songs it contains.
- To avoid an excessively sparse co-occurrence matrix, we sampled playlists containing at least one of **10** songs that popular across playlists.
- We used a **80/10/10%** split for train/val/test.

Methods

- **Global Co-Occurrence Matrix:** We constructed a co-occurrence matrix M based on the presence of song pairs in playlists.
- **High Co-Occurrence Scaling:** With precedent from GloVe, we included a scaling term to reduce the weight of frequently co-occurring songs:

$$\lambda_{(i,j)} \leftarrow \left(\frac{\lambda_{(i,j)}}{\lambda_{\max}}\right)^{\alpha}$$

- **Laplace Smoothing:** We reduced the sparsity of the co-occurrence matrix by incrementing co-occurrence values by 1.
- **Optimization:** We used batch gradient descent to minimize the following objective function by taking its gradient:

$$J = \sum_{i,j}^{n} \left(\frac{M_{(i,j)}}{M_{\max}}\right)^{\alpha} (w_i^T w_j + b_i + b_j + \log M_{(i,j)})^2$$

- **Artist Playlist Augmentation:** To account for musical similarity from artists, we augmented our datasets with "artist playlists" as:

$$\left(M_{(i,j)} \leftarrow M_{(i,j)} + 1\right) \text{ if Artist}(i) = \text{Artist}(j) \text{ and } i \neq j, \forall i, j$$

- **Contrastive Learning with NLP:** Using BERT embeddings, we made a semantic embedding for playlist titles and used playlist similarity to generate positive and negative examples for contrastive learning:

$$J = \sum_{\{(w_i, w_j, y)\}} y \cdot \left(1 - \frac{w_i^T w_j}{\|w_i\|_2 \cdot \|w_j\|_2}\right) + (1-y) \cdot \max\left(0, \frac{w_i^T w_j}{\|w_i\|_2 \cdot \|w_j\|_2} - m\right)$$

Experiments, Discussion, and Future Work

- We trained four models, each with Laplace smoothing:
  - (1) MuSE
  - (2) MuSE with Artist Augmentation
  - (3) MuSE with Contrastive NLP Learning
  - (4) MuSE with Artist Augmentation + Contrastive NLP Learning
- With grid search on the validation set, we trained each model with **150** epochs, learning rate **0.05**, and embedding dimension **500**.
- Model **(4)** gave the best results, with a **81.82% F1 score** and **coherent clustering** sorted by artists and genre.
- These results show the effectiveness of the **MuSE** method, which recovers semantic embeddings without looking at audio features.
- Future work should focus on helping MuSE generalize to unseen songs, exploring methods like supplementary neural networks.

Results

- **Quantitative Evaluation:** We assessed the accuracy, precision, recall, and F1 score for each set of embeddings used on a downstream classification task determining whether labeled song pairs are similar or dissimilar.

Model	Accuracy	Precision	Recall	F1
MuSE	65.30%	71.79%	82.35%	76.71%
MuSE + Artist Augmentation	72.22%	77.92%	82.19%	79.98%
MuSE + Contrastive Learning	69.61%	74.07%	**85.71%**	79.47%
MuSE + Augmentation + Contrastive	**74.55%**	**79.75%**	84.00%	**81.82%**

- **Qualitative Evaluation:** We used t-SNE to reduce the dimensionality of popular songs, and visually assessed the clustering effects of the songs as they pertain to artists, genre, and style.